“Modeling large-scale networks using virtual machines and physical appliances”

by Joseph Mayes, Software Engineering Institute, Carnegie Mellon University

When I began teaching technology courses in the 1990s, the size of your lab and the amount of equipment available to students indicated, if not directly measured, program excellence. Having labs filled with hardware was impressive, and the right amount of hardware was always “more.” Classes were full, and students were coming to us in droves.

And then the IT bubble burst. Interest in training peaked and then crashed. There wasn’t enough money to refresh the hardware labs, and some of the labs didn’t deserve to be refreshed. With fewer students, schools could get by on less equipment.

Also, as technology was maturing, remote teaching was coming of age: students were asking the schools to come to them over the Internet. Citrix, the RDP, and related products allowed labs to be controlled remotely, and VMware had released its first x86 server virtualization product. The meaning of “hands-on training” was changing to include virtual hands-on experiences.

Adapting to Support Remote Training

A team at Carnegie Mellon University’s Software Engineering Institute (SEI) was creating IT security courses for elements of the U.S. Army and Marine Corps through SEI’s affiliation with the U.S. government as a Federally-Funded Research and Development Center (FFRDC) [1]. The courses included PowerPoint lectures, instructor-led demonstrations, and hands-on interactive labs where students operated PCs, servers, routers, firewalls, and other IT systems in an isolated, closed network. The labs generally presented a network in a given start state, with instructions for manipulating the network to reinforce the training objective. As such, the labs were running live operating systems (OSes) and generating real traffic (as opposed to simulations).

The Department of Defense (DoD) contract with the SEI required that students be able to access the training online, on demand, any time, and from any place with an Internet connection. This included delivery of the lecture content and slides, and also a way for students to execute a course’s interactive lab assignments over the Internet.

This presented a number of challenges. The courses, including labs, had to be available around the clock every day. The training delivery connection had to use only standard ports allowed through DoD firewalls (generally restricted to HTTP and HTTPS). The lab solution had to be highly scalable to support a percentage of the more than 50,000 users who were signed up for the online training. The lab solution had to deliver the same labs done in a face-to-face class; because they were designed to be interactive, the labs couldn’t be just videos. The courses had to be accessible from DoD computers, which were managed systems whose end users couldn’t install supplemental software. Because of the restriction above, and also because of restricted bandwidth at some DoD locations, the labs had to operate remotely as opposed to being downloaded and run locally. The lab solution couldn’t be based on ActiveX because the military

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
27 JAN 2014	**N/A**	**-**

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
Modeling large-scale networks using virtual machines and physical appliances	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
Joe Mayes	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)	8. PERFORMING ORGANIZATION REPORT NUMBER
Software Engineering Institute Carnegie Mellon University Pittsburgh, PA 15213	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release, distribution unlimited

13. SUPPLEMENTARY NOTES
The original document contains color images.

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			
unclassified	**unclassified**	**unclassified**	**SAR**	**18**	

disallowed ActiveX support on its systems, which made running an RDP client over ActiveX not possible.

The challenges the SEI encountered in delivering the instruction were markedly different than those it faced in delivering the remote labs.
Instruction delivery had to occur through browser-based web content. The eventual product was a Flash-enabled video screen with a parallel slide presentation in sync with the speaker, as well as a synchronized scrolling text display with accompanying downloadable Portable Document Files (PDFs).
The lab delivery would use a web-enabled portal system that would serve as a proxy to the closed network and system access required to run the lab assignments. Using the portal would allow the RDP, Telnet, SSH, and other protocols to be used in the lab environment without having to cross the DoD firewalls.

The final hurdle was the development of the delivery platform for the lab systems behind the portal. Building dynamically scalable labs would be prohibitively expensive if physical machines were used for all lab components. Also, after a student completed a lab, or if the student crashed the lab in mid-execution, there needed to be a way to bring the lab environment back to a pristine state for its next use. Because each course contained multiple labs, scalability demanded the ability to support multiple lab setups on the same physical equipment to keep the physical footprint to a reasonable size, and selection of the resources and the plumbing of those resources had to be scriptable and automated for each lab.

Developing a Custom Virtualization Solution

Server virtualization offered a number of attractive properties for class labs, both for students in classrooms and online. A single hardware instance could run multiple virtual machines (VMs). 'Snapshots' (the process of capturing the VM at a certain state), allowed the lab to be started with VMs that had a repeatable initial configuration and then launched on demand in any running or non-running state.

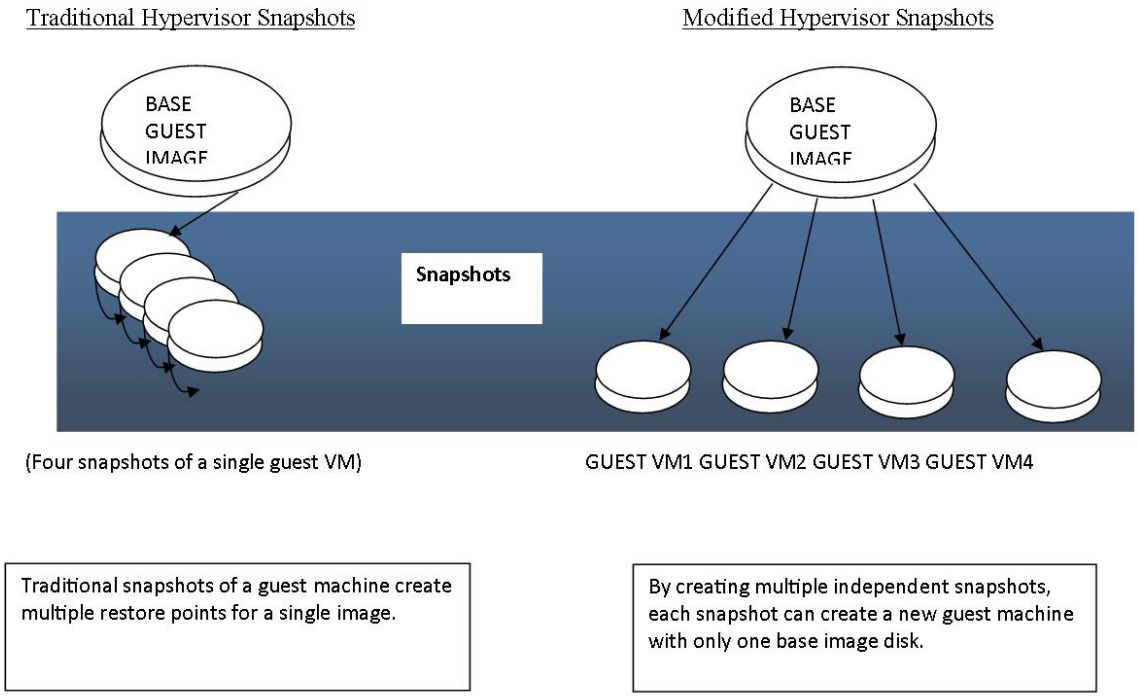
Those same snapshots enabled the capture of the lab VMs at multiple points in time in the lab process. Long labs could be broken into smaller components, and it would no longer be necessary to complete long, multistep labs in a single marathon session or repeat an entire lab if the student incorrectly executed a single step. And after a student completed a lab, the virtual machine(s) could be brought down and reverted back to the initial settings to restart the lab again for the next instance.

While the above virtualization solutions were obvious ways to leverage hypervisor technologies for lab support, resolving the scalability issues took some customization of the hypervisor solution. For instance, while virtualization has obvious advantages for reducing physical hardware needs, it doesn't really address the storage requirements for tens or hundreds (or thousands!) of virtual machines. The solution to that issue was manipulation of the hypervisor snapshot process.

A normal hypervisor VM begins as a base disk of the installed guest OS, with snapshots capturing the differences between the base disk and the VM at its current state, then repeating that process for the next guest OS, and so on. Using this VMware storage model eventually depletes the available space in the VMware File System (VMFS) volume, and because volume size is capped at 2 terabytes, it isn't long before the entire storage capacity is consumed.

There are some generally supported methods for overcoming this issue, but they didn't give the degree of flexibility the SEI team desired. Instead, the team modified the snapshot process by creating a single base disk (of, say, Windows 2008r2 Server). From this base disk, they could then install the system as a domain controller and snap the disk again. The SEI team used a single base disk and created a snapshot of a domain controller, then reused the same base disk to create an Exchange server, then reused the disk again to create a web server, and so on (Figure 1).

Figure 1: Manipulating snapshots for creating VMs



The key to this system, then, was to launch multiple Windows servers by threading the different snapshot images all back to the same disk, which meant that a file server, domain controller, Exchange server, SQL server, and other servers could each be run from one base disk plus the snapshot file for that server.

The same process is also used to clone multiple Windows XP client machines in a network environment. By scripting the launch of XP clients, which is a documented XP capability, it's possible to produce multiple independent XP clients in an environment and join them to a domain, all from a single XP base disk, without ever having to manually install the individual workstations. The process also works for multiple domain controllers or any other system with an automated installation process.

The net result of this work is the ability to store a few base disks and installation automation scripts and create a startup sequence that can spin up an entire exercise environment in just a couple of minutes.

It also resolved the hardware reusability challenge. One could now store all the scripts necessary for multiple labs, select which scripts to invoke, start any lab at will, and then tear it down and have a totally different lab up and running within minutes.

Ultimately, the SEI developed this capability into a training delivery platform called Virtual Training Environment (VTE), which is still used today by the U.S. government as FedVTE [2] and by the SEI as the CERT® Simulation, Training and Exercise Platform--forward (STEPfwd) [3].

Testing the Bounds of Scalability

But could this concept be expanded even further? How could big could one of these virtual environments get?

The question of expansion arose from requests for network environments where real-world scenarios could be played out safely—isolated environments that could hold tens or hundreds of systems and associated networking resources and that could mirror a real-world environment of a business, a government agency, or a military network. This meant taking the lessons learned from VTE lab virtualization and giving that technology a life of its own. The SEI took this idea and developed an exercise network. Previously, virtual machine labs had all been created on a single server, but moving to larger and larger environments required linking multiple hypervisor servers together into one massive virtual environment.

This method introduced new obstacles when it came to multi-server environments and disk storage limitations.

Passing hypervisor Virtual Local Area Network (VLAN) information between physical servers was easily resolved by employing 802.1q VLANs between servers, but the VLAN trunks needed to have enough bandwidth to not be choke points for what was being represented to exercise participants as adjacent physical machines on the same network segment.

The problem of creating hundreds of separate VMs had been solved through the use of base disks plus snapshot images, but now a method was needed to instantiate this across multiple server platforms. At this point, another issue arose: The VMFS file system allowed only 8 simultaneous accesses to the base disk (file), limiting the ability to share the same base disk with multiple servers. One solution to the VMFS issue was merely to create multiple base disks, but limiting the number of base disks while also maintaining multiple mappings for different exercises became a monumental management problem, and keeping changes synchronized would create additional complexities.

The second method was to let a hardware solution reduce the required storage by deduplicating data. The SEI tested and successfully used NetApp storage systems for this purpose. NetApp data deduplication was highly effective for creating multiple instances of a single base disk without increasing the overall storage requirements because each subsequent base disk was a clone of the first one. And the NetApp volume cloning option made it simple to replicate exercise environments when the SEI began supporting large symposia or competitions, which required multiple instances of the same exercise environment. Since the entire exercise could be built on a volume using the base-disk model to save space, the NetApp volume cloning option was used to create large numbers of environments by simply cloning the exercise volume as

many times as necessary to serve as many as 150 or more simultaneous participants for some applications.

The system expansion also drove a file system review. With VMware as the chosen hypervisor product, the available file system choices were directly attached VMFS or external Network File System (NFS) mounts. While the SEI was developing its exercise network, 1Gbps Ethernet connectivity was the highest transport speed deployed, so a VMFS over Fibre Channel (at 4 Gbps) provided higher speed connections for loading the shared base disks concurrently to multiple hypervisors. However, VMFS' limitations of a maximum for volume size and concurrent accesses to the common base disk limited the scalability of this option. Over time, the availability of affordable 10Gbps connections and faster NFS server front ends permitted the SEI to switch to NFS, which is the file system they currently use and which eliminates both the storage size limitation and the concurrent disk access restriction.

Using a hypervisor infrastructure for launching, managing, and decommissioning temporary environments with large numbers of guest machines doesn't fit the normal profile of hypervisor management systems, so the SEI built its own management system. Key features of that system included the ability to launch and retire large numbers of systems concurrently. Scripting also allowed the SEI hypervisor front end to create VM guest systems from scripted startups as opposed to requiring pre-existing VM guest images (used for spawning hundreds of user workstations in an environment, for example), create multiple user accounts and assign them to individual systems and domains, and create many LAN environments while concurrently assigning guest system NICs to those environments.

This article does not specifically discuss the management system and client access systems. But together they generate a web-based user environment portal that gives users access to a console image of one or more virtual systems, chat sessions between members of a team managing systems, the ability to stop and restart systems, and the ability to tile their screen to manage multiple systems concurrently. The management system also allows overseers, such as instructors and exercise white cell personnel, to view a console image of any system or systems across the entire environment, either in real time or through video capture. This allowed overseers performing post-event analysis of the environment and participants to watch what happened on targeted guest systems in the environment [4].

Increasing Realism by re-introducing Physical Devices

By 2011, the SEI had been using its simulation platform for three years in support of classroom instruction, skills evaluation, single- and multi-site cyber exercises and research projects. The verisimilitude between the virtual environment and operational networks had always had some limitations, but they were mainly related to emulations of physical network appliances such as routers and firewalls (where generic Linux systems had been configured to emulate the functions).

The requirement to increase the realism of the exercise environment came to a head with a project for the newly formed U.S. Cyber Command (USCYBERCOM). The military has a

requirement to “Train As We Fight,” a requirement that applied to the cyber exercises the SEI was to create for them.

In the summer of 2011, USCYBERCOM entered into an agreement with the SEI for the construction of a large-scale exercise environment for a military exercise dubbed Cyber Flag. The exercise, the first of its kind for USCYBERCOM, would involve approximately 300 military cybersecurity experts in force-on force actions in an isolated environment where all four U.S. military services could practice their cyber defense skills [5].

Deploying 8000-plus VMs for the DoD

Pure size was one issue. Creating an accurate, real-world environment without interfacing with the real world would first require a robust network of VMs. The SEI initially envisioned that the network would have 5,000 VMs, a number that would eventually grow to 150+ servers supporting more than 8,000 VMs and associated servers. This meant customizing all aspects of the exercise environment, from OS versions to Windows Active Directory to email structures.

The second requirement was to build the network environment to simulate the actual world as closely as possible, to facilitate cybersecurity activities at the network level as well as at the client and server OS levels. This entailed the inclusion of physical network appliances to the exercise network and STEPfwd virtualization environments—in the game, as opposed to the appliances just being part of the invisible support infrastructure. The addition of physical devices would enhance realism by including physical system vulnerabilities such as hardware resets, supporting requirements for data throughput that weren’t possible with the available virtual machines, and placing inline appliances (intrusion detection systems and intrusion prevention systems, or IDS/IPS, for example) in the exercise data streams to meet USCYBERCOM’s design requirements for the environment.

Specifically, USCYBERCOM required that the virtual environment include commercial network appliances from at least five different vendors. While some vendors could provide virtual machine versions of their products, others could only provide physical appliances, so the SEI team began developing solutions for integrating physical devices into the virtual network.

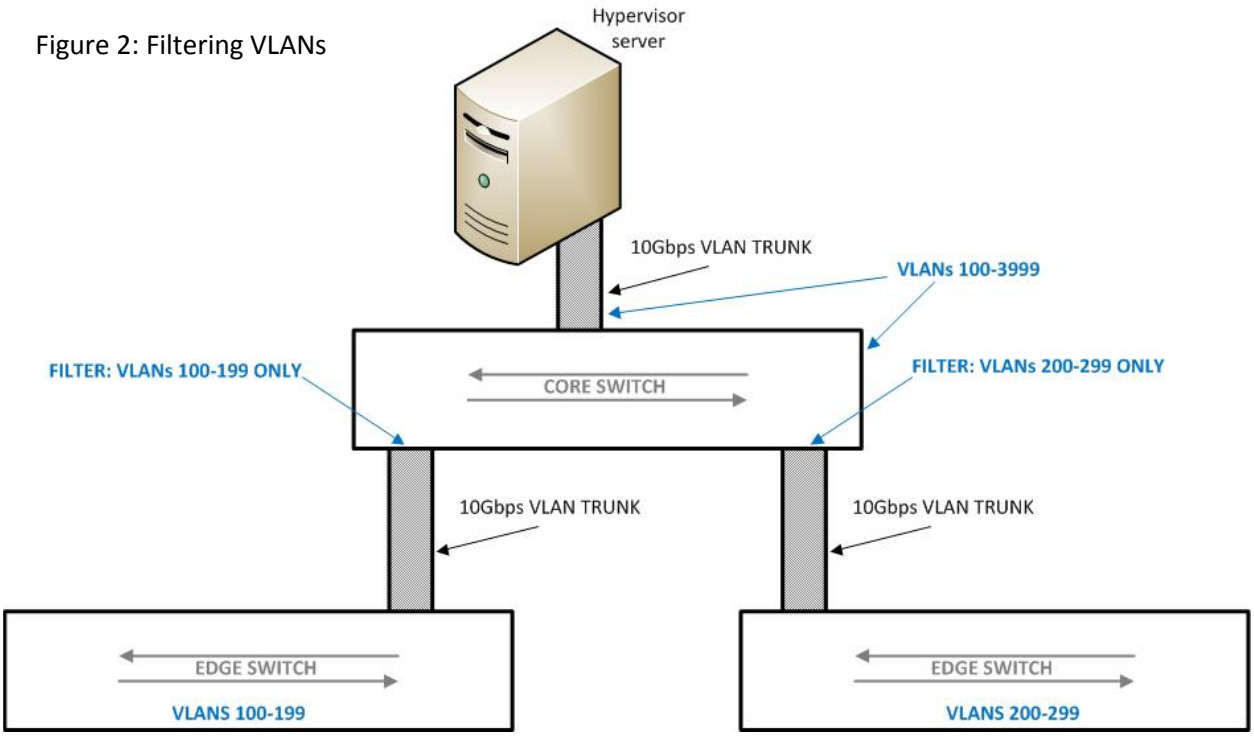
The initial architecture was obvious: 802.1q VLANs and trunks had been used to extend connectivity between physical servers, so they could also be used to extend connectivity to physical network appliances. However, the SEI soon encountered a number of complicating factors.

Virtual devices offer, without adding cost, as many interfaces as the environment requires, but that is not the case for physical environments. Budgets were an issue, so the SEI sometimes had to tradeoff between 100 percent hardware fidelity (the same device model and ports as an enterprise-level production appliance) and a similar device running the same operating system at a lower performance level.

Using VLANs to support connections to physical devices required a large number of VLANs. The largest USCYBERCOM implementation used more than 2,500 VLANs, requiring backbone

infrastructure hardware that supported extended VLAN ranges beyond the first 1,024 VLAN IDs. The large number of VLANs in use also impacted the topology of the exercise. Relatively moderate edge switches could support the appliance requirements, but had limitations on the maximum number of VLANs that could be supported. To ensure the switches did not exceed their 250-VLAN maximum, VLANs were filtered at the trunk boundary so the core switches never presented more than the required VLANs to the edge switches (Figure 2). This was a practical solution that limited the number of VLANs the edge switches had to support, while also providing exercise infrastructure security by denying an aggressor an attack vector via a compromised appliance on one edge switch, which could be used to leverage VLAN hopping to attack systems in another part of the network.

Figure 2: Filtering VLANs



VLANs were filtered between the core switch and edge switches to permit only the required VLANs to propagate to the edge switches.

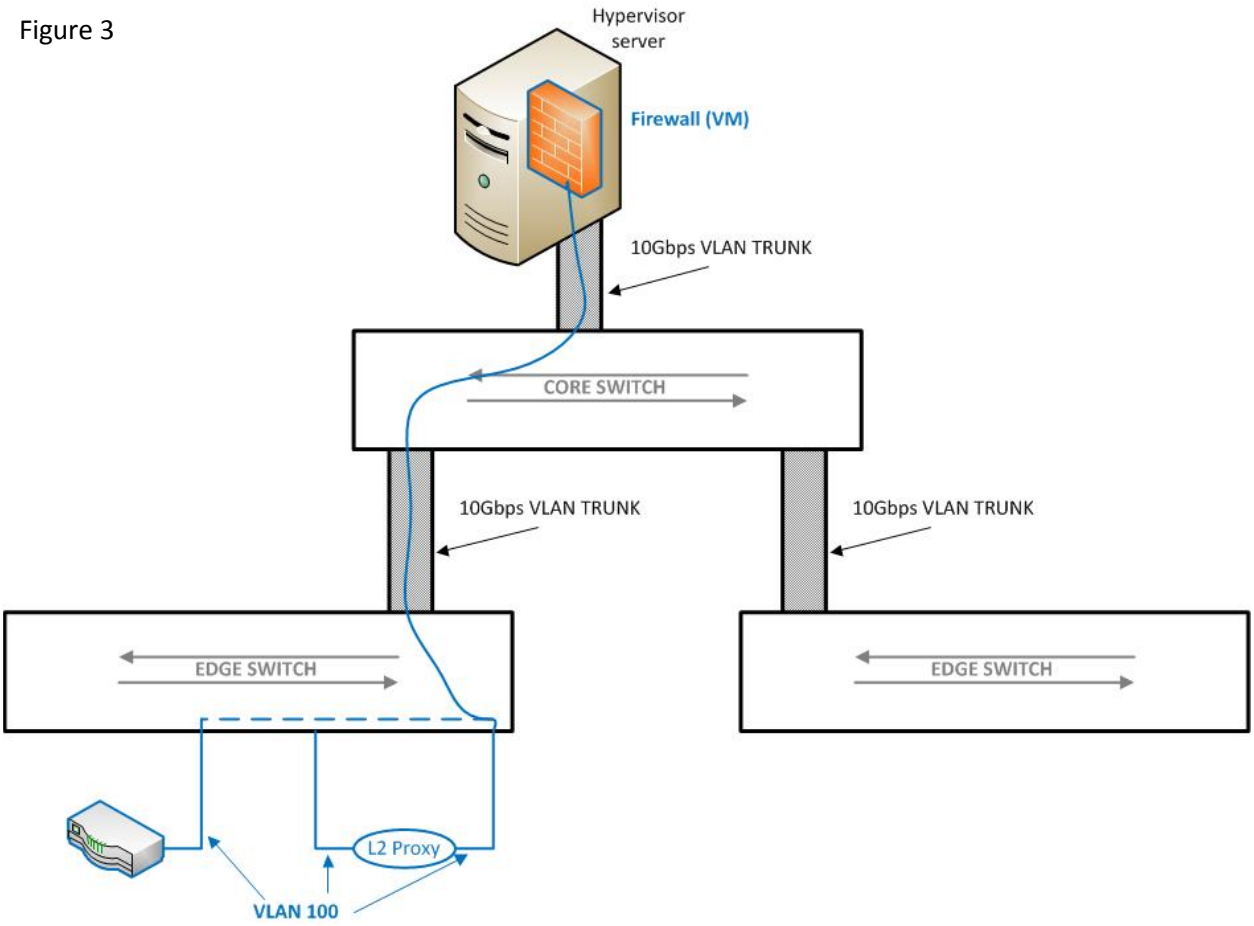
Networking challenges in appliance integration

Making connections between virtual machines and physical appliances presented challenges based on the behaviors of Ethernet and IP. In particular, the behavior of IP address resolution Protocol (ARP) and system ARP tables, and Ethernet Media Access Control (MAC) address tables greatly influenced network design, which had to account for the default behavior of Ethernet physical address resolution.

For example, to meet a networking requirement (as shown in Figure 3), a transparent web proxy appliance was placed in line between two routers connected on the same VLAN (VLAN 100). In a physical environment, this is straightforward, but not so in a logical environment where systems rely on VLAN trunks to support connectivity between physical and logical devices. Initially traffic wouldn't flow through the proxy appliance, as the ARP broadcasts from the router were being seen by the virtual machine firewall in the hypervisor, and switches were passing traffic directly to the MAC address of the virtual firewall and bypassing the web proxy.

Figure 3: Traffic bypassing an in line hardware appliance

Figure 3



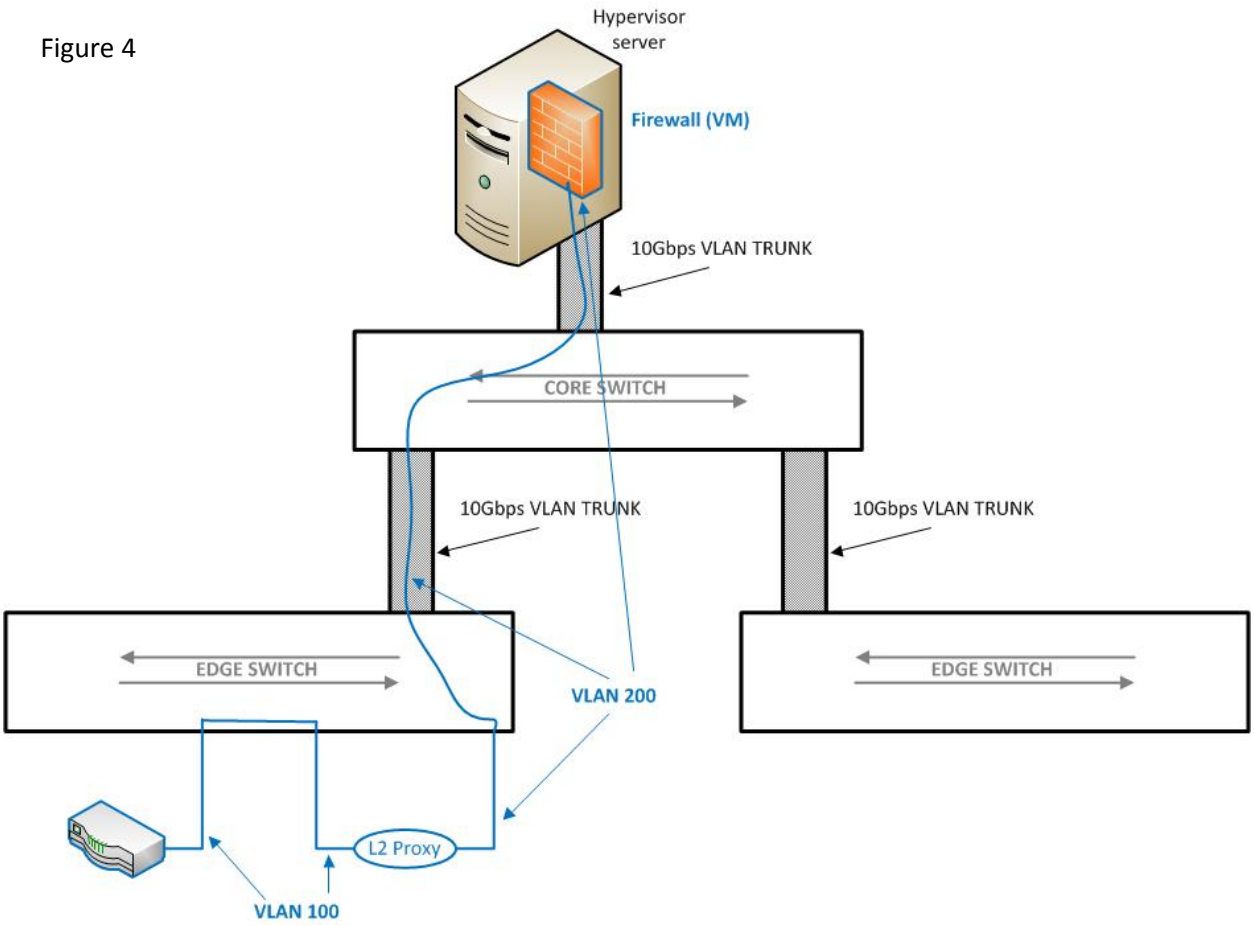
Inline Layer 2 device issue

When the router sends an ARP broadcast for its Ethernet destination MAC address, the router finds the destination MAC on the firewall VM in the hypervisor. Since the router, the proxy and the firewall are all on the same VLAN, the traffic bypasses the proxy and is delivered directly to the firewall.

The solution was to break the pathway into two VLANs as shown in Figure 4 (VLAN 100 and VLAN 200) but using the same single IP subnet on both VLANs. By crossing the IP subnet

through two different VLANs, the ARP request received no reply within its own VLAN (VLAN 100 in the diagram). But because the broadcast also passed through the proxy to VLAN 200, the firewall interface did reply on VLAN 200 and was registered by the switch as an available MAC address through the VLAN 100 port connected to the proxy appliance.

Figure 4



Using two VLANs to control ARP traffic

VLAN 100 connects the router to the proxy's first interface, while VLAN 200 connects the proxy's second interface to the firewall. When the router broadcasts an ARP request for the firewall, the destination MAC address is not found on VLAN 100. The proxy, however, passes the ARP request through its interfaces from VLAN 100 to VLAN 200, where the firewall is discovered, which updates MAC address tables on all devices to reflect MAC address locations. The ARP response returns to the edge switch and registers the firewall MAC as available through the proxy switch ports. Data follows the same path.

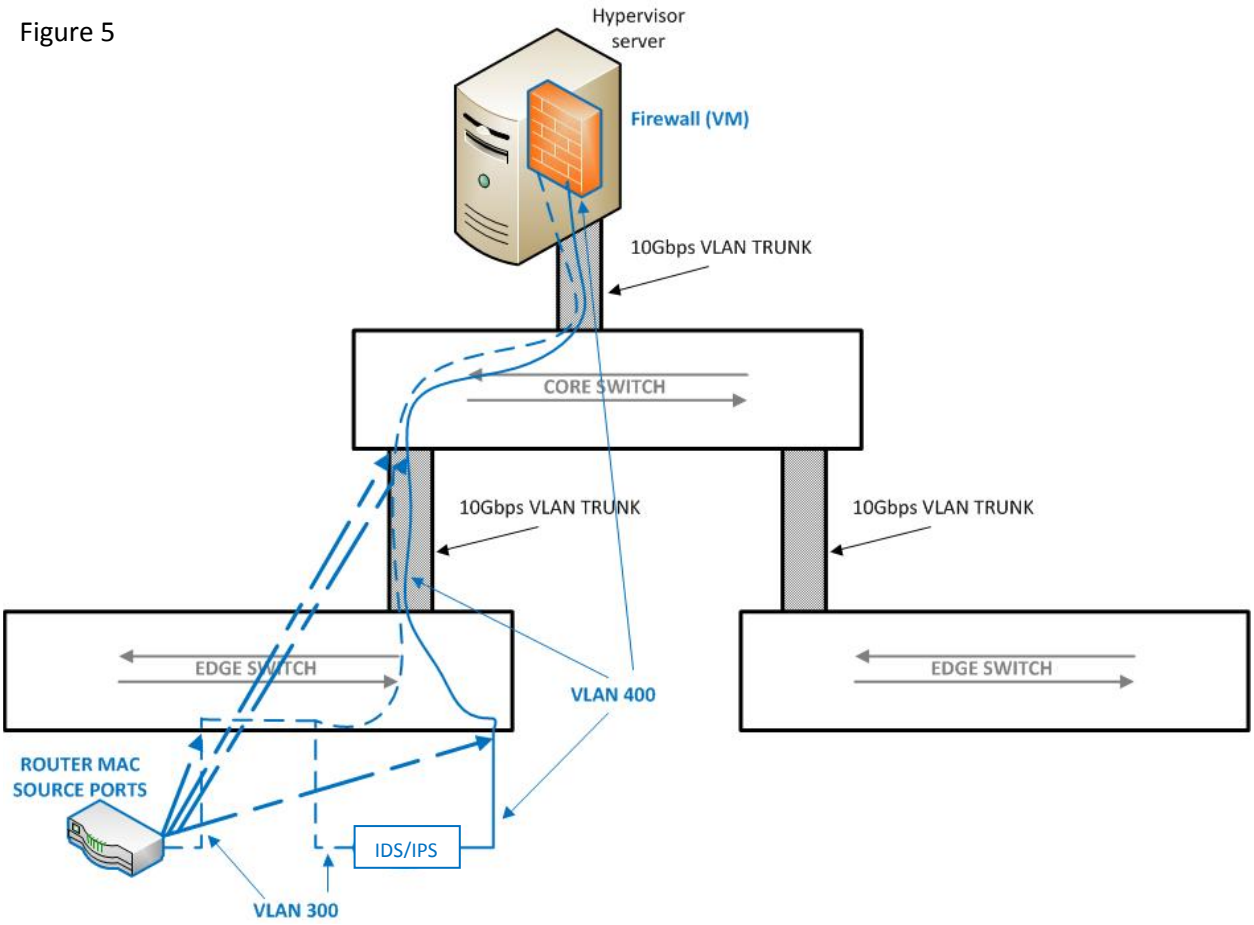
Another challenge was how to place Layer 1 appliances which do not interact with MAC address tables 1. These so-called "bump-in-the-wire" devices do not support the ARP process as

'normal' devices would. In fact, the devices being placed in the network in this instance were Intrusion Detection System/Intrusion Protection System (IDS/IPS) appliances, which are specifically engineered to be 'unseen' on the network.

When an IDS/IPS device was inserted inline between a virtual firewall and a router, the SEI already knew that this connection would require two VLANs. However, the IDS/IPS appliance presented a new problem because the IPS device, as a transparent Layer 1 appliance, appeared to the network as a direct cable connection between two switchports. This connection created multiple entries in the switch MAC address table, because the IDS/IPS device does not keep an ARP table identifying which interface was the source port for the ARP broadcast (figure 5).

When communication began, the switch saw unicast traffic requesting a MAC address that it identified as being sourced from two different VLANs. Both VLANs were reachable from the source appliance, so traffic was likely to end up on either VLAN, causing intermittent communication (as the core switch could gain and then lose the location of the destination and/or source MAC address).

Figure 5

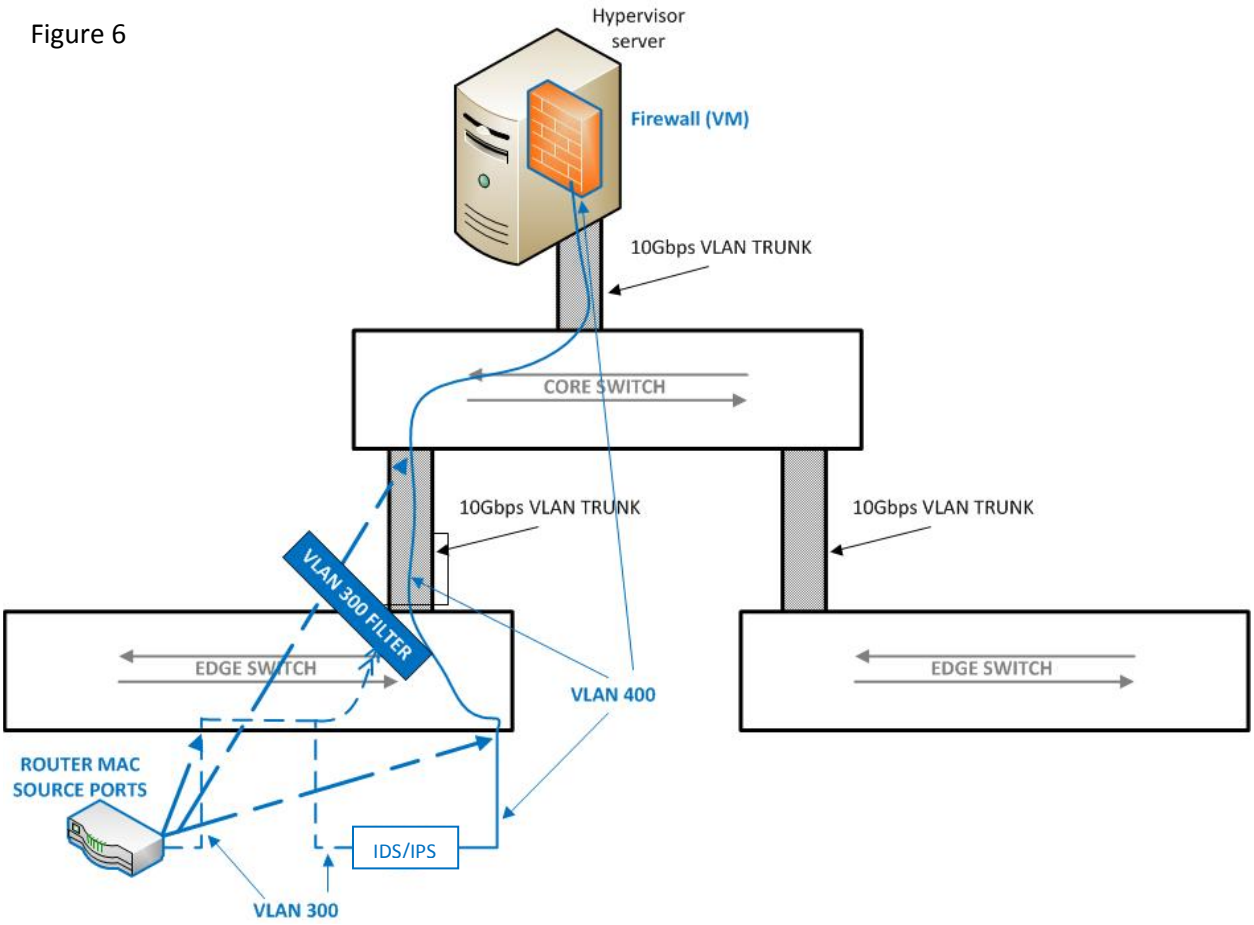


MAC address table confusion from Layer 1 appliances

When the router issues an ARP broadcast, the switch updates the MAC address table to associate the router to its connected interface in VLAN 300. The broadcast then passes through the IDS/IPS, where the switch now sees the router MAC address being re-associated to a second switchport on the same switch (in VLAN 400). When the ARP broadcast reaches the firewall, the ARP reply is also associated to two different ports on two different VLANs. This creates unpredictable delivery behavior for packets attempting to travel between the router and the firewall.

The traffic issue was resolved by only using two VLANs while also filtering the VLANs so that only the VLAN required to reach the next hop destination was visible in the core switch. The strategic point to filter was at the VLAN trunk uplink port from the edge switch. Since the default behavior of a switch is to propagate a broadcast out all ports *except the port from which the broadcast frame was originally received*, the switch automatically guards against Ethernet loops. And since one VLAN was only needed for the connection on the local switch (VLAN 300 in figure 5), blocking that VLAN at the uplink port insured the core switch would learn of the router on VLAN 400, insuring the firewall places its own ARP reply packet on VLAN 400 where it gets passed through the IDS/IPS back to the router in VLAN 300 (figure 6).

Figure 6

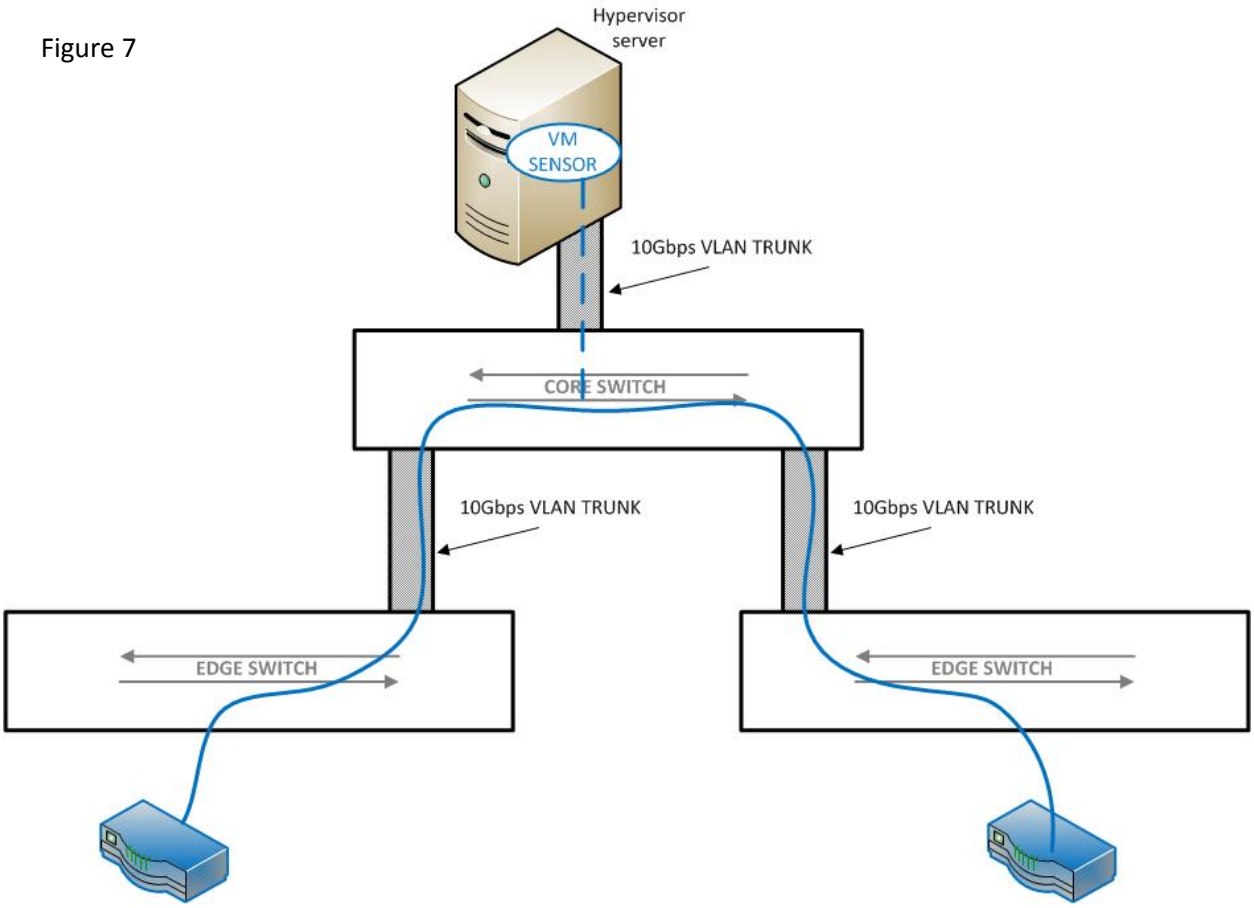


Controlling Layer 1 connections using VLAN filters

By filtering VLAN 300 at the uplink port of the edge switch, the core switch only saw ARP replies in VLAN 400, which then registered MAC address table entries appropriately to permit the correct traffic flow from the core switch to the edge switch, where the traffic was able to move from VLAN 400 to VLAN 300 via the IDS/IPS.

Another networking problem involved the issue of inserting VM sensor systems between two Layer 3 devices (wanting to place a monitoring machine between an edge firewall and an interior router, for example). In this case, Figure 7 shows what happens when trying to expose the sensor to the traffic path: since the source and destination physical appliances see each other in the switched environment, the traffic is never transmitted up the VLAN trunk to the VM sensor device. This happens whether the two appliances are on the same edge switch or whether the appliances ore on different edge switches; the only difference is whether the traffic stays in the edge switch or whether it crosses the core switch before reaching the destination appliance.

Figure 7



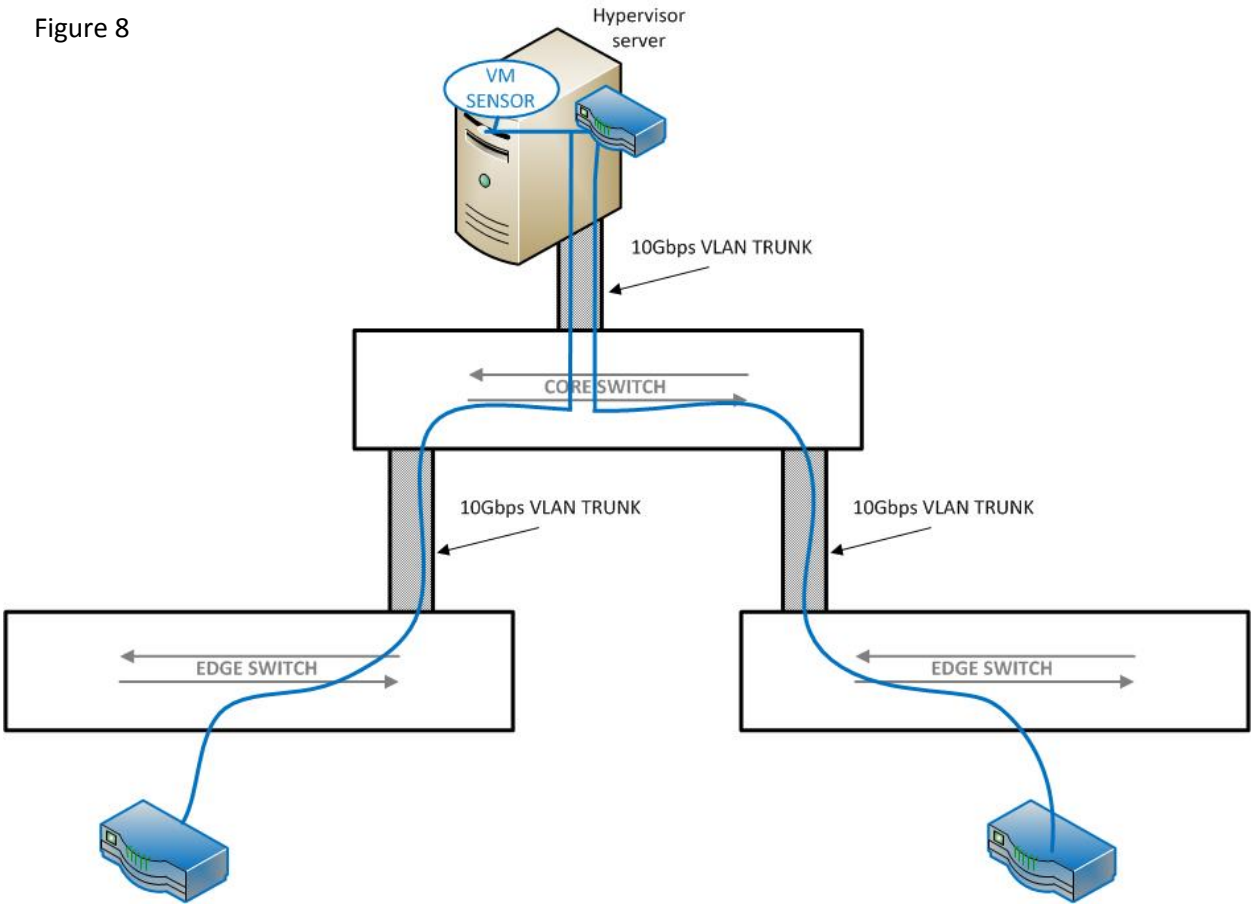
VM sensor system unable to record traffic between physical appliances

Since both physical devices are directly connected to switches, the switches can move traffic through the infrastructure without having the data exposed to the VM sensor.

There was no easy solution to this situation when using ESX VM switches, which cannot be bridged. The SEI ended up creating an artificial Layer 3 device on the VM switch, with the switch placed in promiscuous mode. This permitted packet sniffing by adding an extra hop in the data path (figure 8). It solved the data traffic path issue, but created another IH 'hop' and was no invisible to end users. Additionally, the solution must be engineered with one router port on the same VMnet such that the data you want to sniff sees the sensor before the router so that all Layer 2 and Layer 3 traffic can be sensed.

The SEI deemed other potential solutions, such as mechanically bridging ESX host physical ports, not sufficiently scalable.

Figure 8



Placing another Layer 3 device (router, etc.) in the VM environment forces data past sensor, but the solution is not invisible to the exercise/training environment.

Future development: Layer 1 switching

The infrastructure the SEI currently deploys for exercise and lab support has a fixed, wired Layer 1 infrastructure, which requires manual intervention when cable connections to or between appliances need to be modified. So while hypervisor-based environments can be changed rather easily from one scenario/deployment to the next, the inability to change physical connections via configuration limits the ability to support scenario changes due to the need to change physical cabling to change the environment. The SEI's next iteration of this system will incorporate Layer 1 switches between in-game network appliances and the backbone infrastructure.

Programmable Layer 1 switching will permit connectivity configurations to be stored, and physical wiring changed, on the fly. In one configuration, a router can be connected directly to a proxy before sending the traffic on to a firewall, while reconfiguring the Layer 1 switch would allow the proxy to be changed or the proxy could be taken out of the network entirely when desired for a given scenario.

Another feature of Layer 1 switching is that most L1 switch manufacturers incorporate a form of data tap in their switches. This tap feature makes it much easier to instrument any connection in the system either for use by exercise participants or for transparent monitoring of exercise events by the event operators, which would provide a new solution for connecting VM sensors to networks.

One consideration of Layer 1 switching is that a full Layer 1 switch implementation raises the port count of the systems dramatically, for two reasons. First, all devices now have two extra ports to connect (the "in" and "out" ports on the Layer 1 switch). Second, while it's rare for devices to use all ports of all devices in an environment, this capability is necessary for implementing a fully configurable environment. Specifically, all possible ports of all devices must connect to the Layer 1 switching fabric so that they are all available for configuration if needed in some future environment. If one assumes that it's unlikely that all appliance ports would ever connect to the switching fabric at the same time, it is possible to ultimately oversubscribe the infrastructure's Layer 2 switches because a certain number of appliances will be directly connected to each other, and a certain number of appliance ports may not be used at all. The acceptable degree of oversubscription is a judgment call, and ultimately the infrastructure should be engineered to easily accommodate additional Layer 2 switchports if required.

Adding switch-level configuration to the VM environment

The SEI exercise environments has been unable to provide Layer 2 configuration manipulation to environment participants in the VM environments. This has been a function of creating the isolated environments, because giving participants the ability to manipulate Layer 2 connections allows them to break the environment and also exposes the 'artificiality' of the STEPfwd infrastructure. It's especially true in virtual-only environments because manipulating Layer 2 connectivity usually means giving access to the hypervisor environment, when ideally the participants shouldn't even know a hypervisor environment exists.

An upcoming research project will explore the feasibility of operating shelled hypervisors: a management hypervisor that sets the exercise infrastructure, with an additional virtualized hypervisor running inside the physical hypervisor installation. Within the virtualized hypervisor, the proposed solution would deploy Cisco 1000v virtual switches that would be given to the exercise participants for command-line access, locking the participants out of the hypervisor environment while permitting them to operate switches within their exercise operational sphere. This could complete the creation of a virtual environment that would mirror the physical environment while retaining all the advantages of a virtual environment for training and exercises.

In the physical appliance arena, it's possible to dedicate certain switches to exercise use and turn over configuration to the exercise participants, though it increases the total required equipment and equipment costs and potentially risks the success of the exercise if the switch becomes totally compromised. But, should both of these Layer 2 changes be made, it should be possible to present a total exercise network as a combination of virtual and physical devices for the STEPfwd environment of the future.

CONCLUSION

Whether they're for technical testing, training or training evaluation, individual and/or team proficiency exercises, or just capture-the-flag competitions, lab environments are essential for developing cyber and cybersecurity skills. And, given the current state of technology, these labs will need to incorporate a combination of virtual and physical components to provide the necessary degree of production-environment realism. If done correctly, building a cost-effective, readily deployable lab environment doesn't necessarily entail a compromise in realism.

This material is based upon work funded and supported by the Department of Defense under Contract No. FA8721-05-C-0003 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center.

DM-0000921

REFERENCES AND FURTHER READING

[1] Software Engineering Institute, "SEI Receives Contract Extension From U.S. Government," Carnegie Mellon University, http://www.sei.cmu.edu/newsitems/contract2010_news.cfm

[2] National Initiative for Cybersecurity Careers and Studies, "Federal Virtual Training Environment (FedVTE) and Federal Cyber Training Events (FedCTE)," http://nics.us-cert.gov/training/fedvte-and-fedcte

[3] Software Engineering Institute, "STEPfwd," Carnegie Mellon University, https://stepfwd.cert.org/vte.lms.web

[4] Software Engineering Institute, "XNET CERT Exercise Network," Carnegie Mellon University, http://xnet.cert.org/content/docs/xnet-trifold-2010.pdf

[5] Montalbano, Elizabeth, "U.S. Cyber Command Practices Defense In Mock Attack," Information Week, November 30, 2011,

http://www.informationweek.com/government/security/us-cyber-command-practices-defense-in-mo/232200508

BIOGRAPHY FOR PUBLICATION

JOSEPH MAYES, B.S., M.Ed., is a member of the technical staff at Carnegie Mellon University's Software Engineering Institute and serves with the CERT Enterprise Workforce Development team. He has been an educator for more than 30 years in secondary, undergraduate, and graduate schools, and also as a commercial course instructor. He has been a network engineer and network security professional since 1990, both as a U.S. Army IT Sergeant Major and in civilian positions with government and commercial network systems, including Critical Infrastructure and Key Resources (CIKR) systems. He is a Microsoft Certified Trainer (MCT), a Cisco Certified Academy Instructor (CCAI) and Cisco Certified Systems Instructor (CCSI), and holds more than 25 certifications in Microsoft systems, routing and switching, network security, wireless technologies, telephony and information assurance. He is a member of the IEEE, ISC(2), and the ETA-I. Email: jmayes@sei.cmu.edu.